CLAIMS:

What is claimed is:

1. A method of autonomically patching computer program code, comprising the steps of:

executing a computer program instruction, wherein the computer program instruction is located at the start of a block of code of an execution sequence;

determining if metadata is associated with the computer program instruction;

if metadata is found, redirecting execution to patch instructions indicated by the metadata;

executing the patch instructions;

returning to an instruction of the execution sequence in the computer program.

2. The method of claim 1, wherein the patch instructions are created during execution of the computer program.

3. The method of claim 1, wherein the patch instructions are created by

copying instructions from the block of code to a new memory location;

modifying the order of the instructions of the block of code; and

populating the metadata with pointer to the patch instructions.

4. The method of claim 1, wherein the metadata is in a form of a memory word.

5. The method of claim 1, wherein the metadata includes a pointer to the patch instructions.

6. The method of claim 5, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.

7. The method of claim 6, wherein the starting address includes at least one of an absolute or offset address.

8. The method of claim 1, further comprising checking a new flag in a machine status register to determine whether code patching functionality is enabled.

9. The method of claim 1, wherein the patch instructions include reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.

11. A data processing system for autonomically patching computer program code, the data processing system comprising:

executing means for executing a computer program instruction, wherein the computer program instruction is located at the start of a block of code of an execution sequence;

determining means for determining if metadata is associated with the computer program instruction;

if metadata is found, redirecting means for redirecting execution to patch instructions indicated by the metadata;

executing means for executing the patch instructions;

returning means for returning to an instruction of the execution sequence in the computer program.

12. The data processing system of claim 11, wherein the patch instructions are created during execution of the computer program.

13. The data processing system of claim 11, wherein the patch instructions are created by

copying instructions from the block of code to a new memory location;

modifying the order of the instructions of the block of code; and

populating the metadata with pointer to the patch instructions.

14. The data processing system of claim 11, wherein the metadata is in a form of a memory word.

15. The data processing system of claim 11, wherein the metadata includes a pointer to the patch instructions.

16. The data processing system of claim 15, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.

17. The data processing system of claim 16, wherein the starting address includes at least one of an absolute or offset address.

18. The data processing system of claim 11, further comprising checking means for checking a new flag in a machine status register to determine whether code patching functionality is enabled.

19. The data processing system of claim 11, wherein the patch instructions include reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.

20. A computer program produced in a computer readable medium for autonomically patching computer program code, the computer program product comprising:

first instructions for executing a computer program instruction, wherein the computer program instruction is located at the start of a block of code of an execution sequence;

second instructions for determining if metadata is associated with the computer program instruction;

if metadata is found, third instructions for redirecting execution to patch instructions indicated by the metadata;

fourth instructions for executing the patch instructions;

fifth instructions for returning to an instruction of the execution sequence in the computer program.

21. The computer program product of claim 20, wherein the patch instructions are created during execution of the computer program.

22. The computer program product of claim 20, wherein the patch instructions are created by

copying instructions from the block of code to a new memory location;

modifying the order of the instructions of the block of code; and

populating the metadata with pointer to the patch instructions.

23. The computer program product of claim 20, wherein the metadata is in a form of a memory word.

24. The computer program product of claim 20, wherein the metadata includes a pointer to the patch instructions.

25. The computer program product of claim 24, wherein the pointer to the patch instructions includes a starting

address of the patch instructions in an allocated memory location.

26. The computer program product of claim 25, wherein the starting address includes at least one of an absolute or offset address.

27. The computer program product of claim 20, further comprising sixth instructions for checking means for checking a new flag in a machine status register to determine whether code patching functionality is enabled.

28. The computer program product of claim 20, wherein the patch instructions include reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.